

Encrypted Network Traffic Classification Using Deep and Parallel Network-In-Network Models

Thota Devaki Sai Priya

PG scholar, Department of MCA, DNR College, Bhimavaram, Andhra Pradesh.

K.Sridevi

(Assistant Professor), Master of Computer Applications, DNR college, Bhimavaram, Andhra Pradesh.

Abstract Now-a-days internet is everywhere and responsible for generating different types of network traffic such as Email traffic, video streaming, browsing and many more. Analysing or predicting different types of traffic can help in knowing which type of traffic is in more usage and which is in less usage. There are many deep and machine learning algorithms available to classify different network types but they lack support of Network in Network model and Global Average Pooling (helps in generating one feature map for each corresponding category of the classification task and this features mapping help in reducing model parameters and better classification accuracy). Deep Network in Network model will use parallel different layers to train Network Packet Header and Packet body whereas existing algorithms were using same layers to train both header and body. This paper presents a neural network model with deep and parallel network-in-network (NIN) structures for classifying encrypted network traffic. Comparing with standard convolutional neural networks (CNN), NIN adopts a micro network after each convolution layer to enhance local modelling. Besides, NIN utilizes a global average pooling instead of traditional fully connected layers before final classification, which reduces the number of model parameters significantly. In this proposed method, deep NIN models with multiple MLP convolutional layers are built to map fixed-length packet vectors towards application or traffic labels.

I. Introduction

With the increasing reliance on the internet for diverse services such as email, video streaming, web browsing, and file transfers, the volume and complexity of network traffic have grown exponentially. Understanding and classifying this traffic, especially when encrypted, is essential for network management, cybersecurity, and ensuring optimal resource utilization. Traditional deep learning methods like standard Convolutional Neural Networks (CNNs) have shown promise in traffic classification tasks but often struggle with extracting distinct features from encrypted data, particularly when treating packet headers and bodies uniformly. These models also suffer from

high parameter counts due to fully connected layers, leading to increased computational cost and reduced scalability.

To address these limitations, the proposed approach introduces a Deep and Parallel Network-In-Network (NIN) model for encrypted network traffic classification. Unlike traditional CNNs, the NIN architecture integrates micro-networks after each convolution layer to enhance feature learning and employs Global Average Pooling to reduce model complexity while maintaining high accuracy. The model processes packet headers and payloads through parallel layers to capture nuanced information from both components independently. Trained on the ISCX VPN-nonVPN dataset, the proposed NIN model demonstrates superior performance over standard CNNs, achieving higher classification accuracy, precision, and robustness in identifying different encrypted traffic types such as email, audio streaming, and file transfers. This approach enhances traffic analysis efficiency while supporting modern network security requirements.

II. Literature Survey

1. Wang, W., Zhu, M., Zeng, X., Ye, X., & Sheng, Y. (2017) – “Malware Traffic Classification Using Convolutional Neural Network for Representation Learning”

This study demonstrated the use of CNNs for automated feature extraction and classification of malware traffic from raw network data. The success of CNNs in learning hierarchical features from packet data motivated further exploration of deep learning for encrypted traffic analysis, highlighting CNNs' ability to handle packet-level classification without manual feature engineering.

2. Lopez-Martin, M., Carro, B., Sanchez-Esguevillas, A., & Lloret, J. (2017) –

“Network Traffic Classification Based on Convolutional Neural Networks”

This paper proposed a CNN-based traffic classifier and confirmed its ability to classify encrypted and non-encrypted traffic effectively. However, the study noted limitations in scalability and model complexity, encouraging the use of lightweight CNN variants or alternate pooling methods like Global Average Pooling to reduce parameters and increase efficiency.

3. **Shafiq, M. Z., Khayam, S. A., & Farooq, M. (2008) – “Embedded Malware Detection Using Markov n-Grams”**

Although based on Markov models, this early work laid the foundation for behavior-based traffic classification. It emphasized the importance of packet header and payload inspection in classification tasks, a concept later adapted in deep learning models like the proposed parallel NIN architecture for separate header-body processing.

4. **Lotfollahi, M., Jafari Siavoshani, M., Shirali Hossein Zade, R., & Saberian, M. (2020) – “Deep Packet: A Novel Approach for Encrypted Traffic Classification Using Deep Learning”**

This study introduced Deep Packet, a deep learning model trained on encrypted traffic datasets. It confirmed that even encrypted payloads contain sufficient information for classification and stressed the importance of using advanced models capable of feature extraction at both spatial and temporal levels—paving the way for NIN-style models.

5. **Lin, M., Chen, Q., & Yan, S. (2013) – “Network in Network”**

This foundational paper proposed the Network in Network (NIN) architecture, which replaces traditional convolution layers with micro neural networks to improve feature abstraction. Its use of global average pooling to replace fully connected layers not only reduced

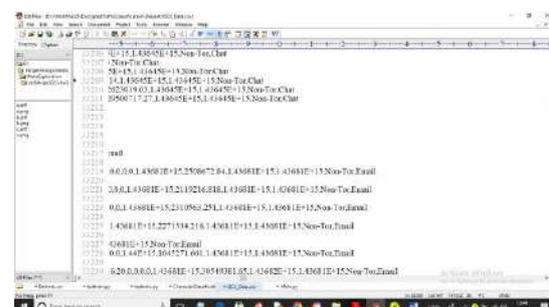
parameters but also improved generalization, making it a key inspiration for your proposed method.

Proposed Method

The proposed method introduces a deep learning framework for classifying encrypted network traffic using a Deep and Parallel Network-In-Network (NIN) architecture. Unlike traditional CNN models that process both packet headers and bodies through the same layers, the proposed model uses parallel processing streams to separately handle these components, allowing for more accurate feature extraction. By incorporating multiple MLP convolutional layers and utilizing Global Average Pooling instead of fully connected layers, the model significantly reduces the number of parameters while improving classification accuracy. Trained on the ISCX VPN-nonVPN dataset, this approach effectively classifies various traffic types such as Email, Chat, Browsing, and Streaming, even in encrypted form, outperforming standard CNNs in accuracy and robustness.

Results

To train propose parallel NIN model author using Encrypted Network dataset called ‘ISCX VPN-nonVPN’ which consists of different traffic. Below screen showing dataset details used to train NIN model



| Time | Source IP | Destination IP | Classification |
|-------|-------------|----------------|----------------|
| 11:07 | 172.17.0.1 | 192.168.1.1 | NewTo:Chat |
| 11:08 | 192.168.1.1 | 192.168.1.1 | NewTo:Chat |
| 11:09 | 192.168.1.1 | 192.168.1.1 | NewTo:Chat |
| 11:10 | 192.168.1.1 | 192.168.1.1 | NewTo:Chat |
| 11:11 | 192.168.1.1 | 192.168.1.1 | NewTo:Chat |
| 11:12 | 192.168.1.1 | 192.168.1.1 | NewTo:Chat |
| 11:13 | 192.168.1.1 | 192.168.1.1 | NewTo:Chat |
| 11:14 | 192.168.1.1 | 192.168.1.1 | NewTo:Chat |
| 11:15 | 192.168.1.1 | 192.168.1.1 | NewTo:Chat |
| 11:16 | 192.168.1.1 | 192.168.1.1 | NewTo:Chat |
| 11:17 | 192.168.1.1 | 192.168.1.1 | NewTo:Chat |
| 11:18 | 192.168.1.1 | 192.168.1.1 | NewTo:Chat |
| 11:19 | 192.168.1.1 | 192.168.1.1 | NewTo:Chat |
| 11:20 | 192.168.1.1 | 192.168.1.1 | NewTo:Chat |
| 11:21 | 192.168.1.1 | 192.168.1.1 | NewTo:Chat |
| 11:22 | 192.168.1.1 | 192.168.1.1 | NewTo:Chat |
| 11:23 | 192.168.1.1 | 192.168.1.1 | NewTo:Chat |
| 11:24 | 192.168.1.1 | 192.168.1.1 | NewTo:Chat |
| 11:25 | 192.168.1.1 | 192.168.1.1 | NewTo:Chat |
| 11:26 | 192.168.1.1 | 192.168.1.1 | NewTo:Chat |
| 11:27 | 192.168.1.1 | 192.168.1.1 | NewTo:Chat |
| 11:28 | 192.168.1.1 | 192.168.1.1 | NewTo:Chat |
| 11:29 | 192.168.1.1 | 192.168.1.1 | NewTo:Chat |
| 11:30 | 192.168.1.1 | 192.168.1.1 | NewTo:Chat |
| 11:31 | 192.168.1.1 | 192.168.1.1 | NewTo:Chat |
| 11:32 | 192.168.1.1 | 192.168.1.1 | NewTo:Chat |
| 11:33 | 192.168.1.1 | 192.168.1.1 | NewTo:Chat |
| 11:34 | 192.168.1.1 | 192.168.1.1 | NewTo:Chat |
| 11:35 | 192.168.1.1 | 192.168.1.1 | NewTo:Chat |
| 11:36 | 192.168.1.1 | 192.168.1.1 | NewTo:Chat |
| 11:37 | 192.168.1.1 | 192.168.1.1 | NewTo:Chat |
| 11:38 | 192.168.1.1 | 192.168.1.1 | NewTo:Chat |
| 11:39 | 192.168.1.1 | 192.168.1.1 | NewTo:Chat |
| 11:40 | 192.168.1.1 | 192.168.1.1 | NewTo:Chat |
| 11:41 | 192.168.1.1 | 192.168.1.1 | NewTo:Chat |
| 11:42 | 192.168.1.1 | 192.168.1.1 | NewTo:Chat |
| 11:43 | 192.168.1.1 | 192.168.1.1 | NewTo:Chat |
| 11:44 | 192.168.1.1 | 192.168.1.1 | NewTo:Chat |
| 11:45 | 192.168.1.1 | 192.168.1.1 | NewTo:Chat |
| 11:46 | 192.168.1.1 | 192.168.1.1 | NewTo:Chat |
| 11:47 | 192.168.1.1 | 192.168.1.1 | NewTo:Chat |
| 11:48 | 192.168.1.1 | 192.168.1.1 | NewTo:Chat |
| 11:49 | 192.168.1.1 | 192.168.1.1 | NewTo:Chat |
| 11:50 | 192.168.1.1 | 192.168.1.1 | NewTo:Chat |
| 11:51 | 192.168.1.1 | 192.168.1.1 | NewTo:Chat |
| 11:52 | 192.168.1.1 | 192.168.1.1 | NewTo:Chat |
| 11:53 | 192.168.1.1 | 192.168.1.1 | NewTo:Chat |
| 11:54 | 192.168.1.1 | 192.168.1.1 | NewTo:Chat |
| 11:55 | 192.168.1.1 | 192.168.1.1 | NewTo:Chat |
| 11:56 | 192.168.1.1 | 192.168.1.1 | NewTo:Chat |
| 11:57 | 192.168.1.1 | 192.168.1.1 | NewTo:Chat |
| 11:58 | 192.168.1.1 | 192.168.1.1 | NewTo:Chat |
| 11:59 | 192.168.1.1 | 192.168.1.1 | NewTo:Chat |
| 12:00 | 192.168.1.1 | 192.168.1.1 | NewTo:Chat |

In above dataset each row contains one traffic data and in last column we have traffic classification labels as Email, Chat, Browsing and etc. so by using above dataset we will train Existing Standard

CNN and propose NIN Parallel CNN and then compare their performance in terms of accuracy.

To implement this project we have designed following modules

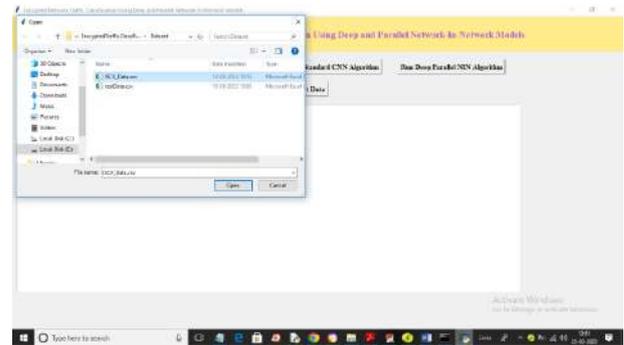
- 1) Upload ISCX VPN-nonVPN Dataset: using this module we will upload dataset to application and then find and plot graph of different traffic found in dataset
- 2) Dataset Preprocessing: using this module we will process dataset to remove missing values, normalization, shuffling and split dataset into train and test where application using 80% dataset for training and 20% for testing
- 3) Run Standard CNN Algorithm: 80% processed data will be input to standard CNN to trained a model and this model will be applied on 20% test data to calculate classification accuracy
- 4) Run Deep Parallel NIN Algorithm: 80% processed data will be input to Deep Parallel NIN CNN to trained a model and this model will be applied on 20% test data to calculate classification accuracy
- 5) Comparison Graph: using this module we will plot accuracy comparison graph between both algorithms
- 6) Traffic Classification using Encrypted Test Data: using this module we will input TEST data and then NIN model will classify test data into possible traffic types.

SCREEN SHOTS

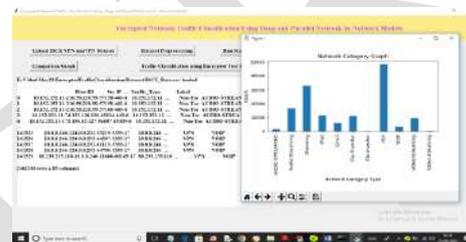
To run project double click on run.bat file to get below screen



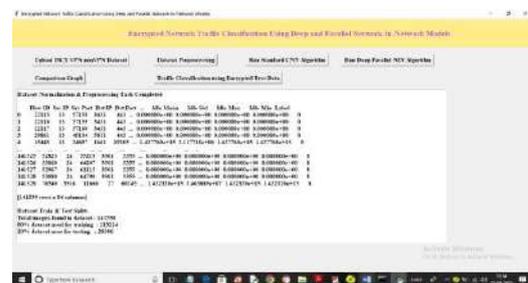
In above screen click on 'Upload ISCX VPN-nonVPN Dataset' button to upload dataset and get below output



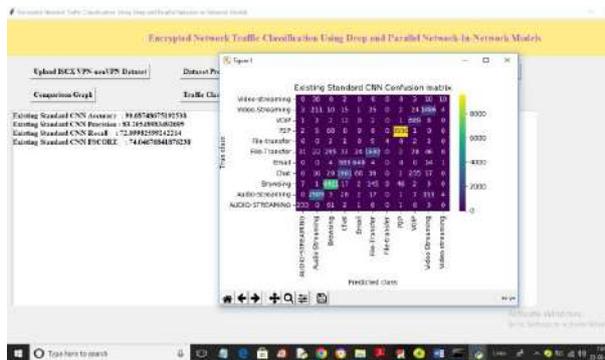
In above screen selecting and uploading dataset and then click on 'Open' button to load dataset and get below output



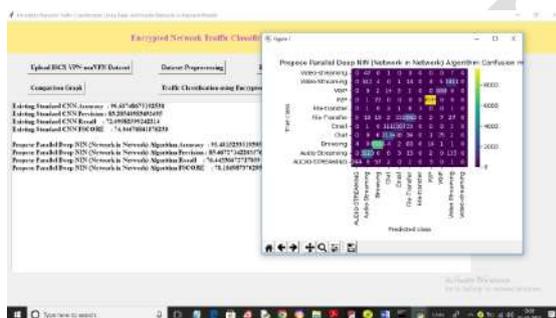
In above screen dataset loaded and we can see dataset contains both numeric and non-numeric data with '.' Symbols as encrypted data but deep learning algorithm only accept numeric data so we need to convert above data into numeric format by applying Label Encoding class. In above graph x-axis represents different traffic types exists in dataset and y-axis represents counts and now close above graph and then click on 'Dataset Preprocessing' button to process dataset and get below output



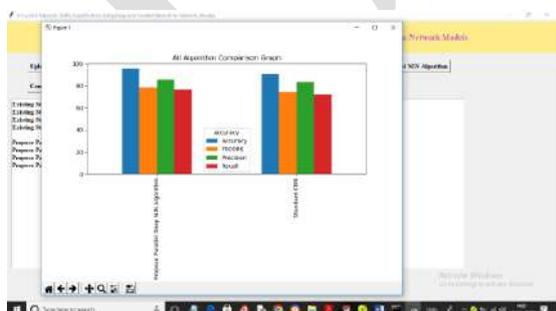
In above screen entire dataset converted to numeric format and then in last lines we can see total records exists in dataset and we can see train and test split details and now click on 'Run Standard CNN Algorithm' to train existing CNN and get below output



In above screen with Existing CNN we got 90% accuracy and we can see other metrics also and in confusion matrix graph x-axis represents Predicted Labels and y-axis represents True Labels and all blue boxes contains INCORRECT prediction count which are few and all different colour boxes contains CORRECT prediction count. Now close above graph and then click on 'Run Deep Parallel NIN Algorithm' button to train propose algorithm and get below output

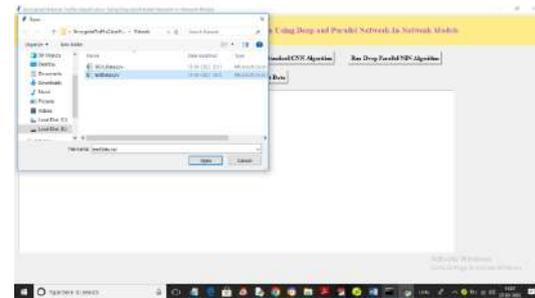


In above screen with Propose Deep Parallel NIN model we got 95% accuracy and we can see confusion matrix graph also and now click on 'Comparison Graph' button to get below graph

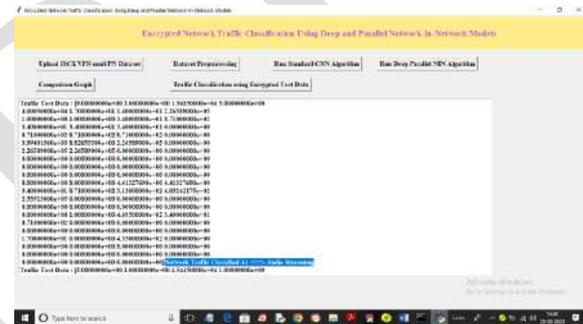


In above graph x-axis represents algorithm names and y-axis represents accuracy, precision and other metrics in different colour bars and in both algorithm propose NIN model got high performance. Now click on 'Traffic Classification

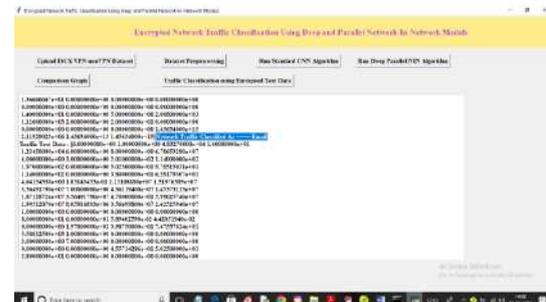
using Encrypted Test Data' button to upload test and then NIN model will classify traffic



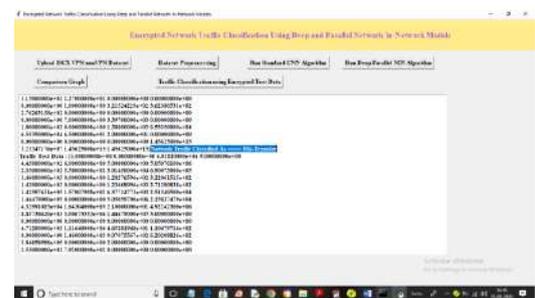
In above screen selecting and uploading Test Data file and this file will not have traffic classification label and NN model will analyse above file and predict traffic type and get below output



In above screen in square bracket we can see test data and then in blue colour text we can see traffic predicted as 'Audio Streaming' and scroll down above screen to view all predicted output

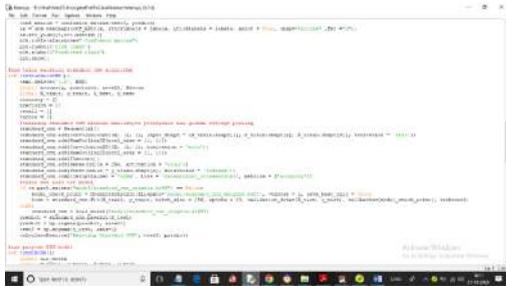


In above screen traffic classified as Email

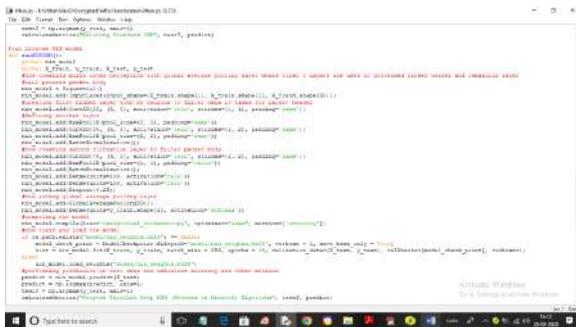


In above screen traffic classified as 'File Transfer'. Similarly by following above screens you can run code

In below screen showing code for Existing Standard CNN



In above screen read red colour comments to know about normal standard CNN



In above screen defining propose parallel NIN model

Conclusion

In conclusion, the proposed Deep and Parallel Network-In-Network (NIN) model provides an efficient and accurate solution for encrypted network traffic classification. By leveraging parallel processing for packet headers and bodies, and incorporating global average pooling to reduce complexity, the model achieves superior performance compared to traditional CNNs. Trained on the ISCX VPN-nonVPN dataset, it demonstrates high classification accuracy across various encrypted traffic types such as email, chat, and streaming. The approach not only enhances detection accuracy but also reduces computational overhead, making it a scalable and practical

solution for modern network traffic analysis in encrypted environments.

References

- [1] W. Wang, M. Zhu, X. Zeng, X. Ye, and Y. Sheng, "Malware traffic classification using convolutional neural network for representation learning," in *Proc. Int. Conf. Information Networking (ICOIN)*, 2017, pp. 712–717.
- [2] M. Lopez-Martin, B. Carro, A. Sanchez-Esguevillas, and J. Lloret, "Network traffic classifier with convolutional and recurrent neural networks for Internet of Things," *IEEE Access*, vol. 5, pp. 18042–18050, 2017.
- [3] M. Z. Shafiq, S. A. Khayam, and M. Farooq, "Embedded malware detection using Markov n-grams," in *Detection of Intrusions and Malware, and Vulnerability Assessment*, Springer, 2008, pp. 88–107.
- [4] M. Lotfollahi, M. Jafari Siavoshani, R. Shirali Hossein Zade, and M. Saberian, "Deep Packet: A novel approach for encrypted traffic classification using deep learning," *Soft Computing*, vol. 24, no. 3, pp. 1999–2012, 2020.
- [5] M. Lin, Q. Chen, and S. Yan, "Network in network," arXiv preprint arXiv:1312.4400, 2013.
- [6] J. Zhang, Y. Chen, and B. Liu, "Network traffic classification using deep learning: A review," *IEEE Access*, vol. 6, pp. 79050–79059, 2018.
- [7] R. Wang, Y. Zeng, and L. Wang, "Encrypted traffic classification based on deep learning with semantic feature vectors," in *Proc. IEEE Int. Conf. Communications Workshops (ICC Workshops)*, 2019, pp. 1–6.
- [8] T. Nguyen and G. Armitage, "A survey of techniques for Internet traffic classification using machine learning," *IEEE Communications Surveys & Tutorials*, vol. 10, no. 4, pp. 56–76, 2008.
- [9] A. Dainotti, A. Pescapé, and K. Claffy, "Issues and future directions in traffic classification," *IEEE Network*, vol. 26, no. 1, pp. 35–40, Jan.-Feb. 2012.
- [10] J. Wang, X. Zhang, and L. Sun, "A survey on encrypted traffic classification," in *Proc. Int. Conf. Computing, Networking and Communications (ICNC)*, 2020, pp. 610–614.